

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

RIEŠENIE OPTIMALIZAČNÝCH ÚLOH PRE ELEKTROMOBILY
DIPLOMOVÁ PRÁCA

2024

BC. MICHAL UHRIN

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

RIEŠENIE OPTIMALIZAČNÝCH ÚLOH PRE ELEKTROMOBILY
DIPLOMOVÁ PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: prof. RNDr. Mária Lucká, PhD.

Bratislava, 2024
Bc. Michal Uhrin



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Michal Uhrin
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Riešenie optimalizačných úloh pre elektromobily
Solving optimization problems for electric vehicle routing problems

Anotácia: Jedným z najväčších producentov emisií CO₂ je automobilová doprava. Sme svedkami celosvetovej snahy o zníženie emisií optimalizáciou dopravy a zavedením elektromobility, ktorej sme v súčasnosti svedkami. Nárast počtu elektrických vozidiel prináša so sebou aj niekoľko nových výziev, akými sú napríklad optimalizácie nabíjania elektromobilov, minimalizácia ich nákladov a pre veľké dopravné spoločnosti aj optimalizácia zásobovacích trás. Riešenie problémov zásobovania veľkými nákladnými elektrickými autami je jednou z úloh, ktorých riešenie je potrebné hľadať pomocou použitia vhodných metód umelej inteligencie. Nakoľko sa jedná o NP-úplný problém, existuje viacero heuristických metód na jeho riešenie. Okrem nájdenia optimálnej trasy je problém navyše komplikovaný nájdením vhodnej dobíjacej stanice a času stráveného dobitím.

Cieľ: Analyzujte problémy súvisiace s použitím nákladných elektromobilov pri rozvoze tovarov. Preskúmajte vhodnosť použitia metaheuristiky a navrhňte vhodné riešenie. Metódu implementujte a porovnajte s existujúcimi riešeniami na dostupných množinách.

Literatúra: Bonilha, I. S., Mavrovouniotis, M., Muller, F. M., Ellinas, G., & Polycarpou, M. (2020). Ant Colony optimization with Heuristic Repair for the Dynamic Vehicle Routing Problem. 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, 313–320. <https://doi.org/10.1109/SSCI47803.2020.9308156>
Mavrovouniotis, M., Li, C., Ellinas, G., & Polycarpou, M. (2022). Solving the Electric Capacitated Vehicle Routing Problem with Cargo Weight. 2022 IEEE Congress on Evolutionary Computation, CEC 2022 - Conference Proceedings, July. <https://doi.org/10.1109/CEC55065.2022.9870383>
Fazeli, S. S., Venkatachalam, S., & Smereka, J. M. (2023). Efficient algorithms for electric vehicles' min-max routing problem. Sustainable Operations and Computers, 0–20. <https://doi.org/10.1016/j.susoc.2023.07.002>
Lezama, F., Soares, J., Faia, R., Vale, Z., Macedo, L. H., & Romero, R. (2019). Business models for flexibility of electric vehicles: Evolutionary computation for a successful implementation. GECCO 2019 Companion - Proceedings of the 2019 Genetic and Evolutionary Computation Conference Companion, December 2020, 1873–1878. <https://doi.org/10.1145/3319619.3326807>

Čestné prehlásenie: Čestne prehlasujem, že som túto prácu vypracovala samostatne iba s použitím uvedených zdrojov a na základe konzultácií so školiteľom a konzultantkou.

.....

Pod'akovanie: Ďakujem mojej kolegini a oficiálnej vedúcej práce Márií Lucká za trpezlivosť a konštruktívne pripomienky k práci a mojej drahej kolegyni/kolega "meno" za odbornú pomoc a didaktické rady.

.....

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Predhovor

Predhovor je všeobecná informácia o práci, obsahuje hlavnú charakteristiku práce a okolnosti jej vzniku. Autor zdôvodní výber témy, stručne informuje o cieľoch a význame práce, spomenie domáci a zahraničný kontext, komu je práca určená, použité metódy, stav poznania; autor stručne charakterizuje svoj prístup a svoje hľadisko.

Obsah

Predhovor	vii
0.1 Motivácia	1
0.2 Cieľ práce	1
0.3 Štruktúra práce	1
0.4 Prínos práce	2
1 Teoretické základy elektromobility a optimalizačných metód	3
2 Prehľad existujúcich prístupov k optimalizácii trás a dobíjania pre elektromobily	5
3 Popis Ant Colony Optimization a ďalších potenciálnych heuristík	7
3.0.1 Úvod	7
3.0.2 Prispôsobenie sa dynamickým zmenám	7
3.0.3 Formulácia problému dynamického trasovania vozidiel (DVRP)	7
3.0.4 Generovanie dynamických testových prostredí	8
4 Implementácia a experimenty, vrátane porovnania výsledkov rôznych metód	9
4.1 Načítanie benchmarkov	9
4.2 Heuristika generovania	10
4.3 Vyhodnotenie	10
5 Diskusia a závery, vrátane návrhov budúcich výskumných smerov	11
Záver	13
Literatúra	15

Zoznam obrázkov

Zoznam tabuliek

Todo list

Úvod

0.1 Motivácia

V súčasnosti stojíme pred obrovskými výzvami v oblasti trvalo udržateľnej mobility a ochrany životného prostredia. S nárastom povedomia o environmentálnych dopadoch tradičných vozidiel s spaľovacími motormi sa zvyšuje dopyt po alternatívnych riešeniach. V tomto kontexte nabíjateľné elektromobily (EVs) predstavujú sľubnú alternatívu, ktorá ponúka nielen redukciu emisií skleníkových plynov a lokálnych emisií, ale aj potenciál na vytvorenie efektívnejšieho a udržateľnejšieho dopravného systému.

Avšak, úspech elektromobility je stále ovplyvnený niekoľkými faktormi, medzi ktoré patria obmedzený dojazd, nedostatok nabíjacích staníc a náklady na elektromobily. Jedným zo spôsobov, ako riešiť tieto problémy, je využitie optimalizačných metód na plánovanie trás a dobíjania pre elektromobily. V tejto práci sa budeme zaoberať práve touto problematikou a budeme sa snažiť nájsť efektívne riešenia pomocou heuristík, s dôrazom na Ant Colony Optimization (ACO) a prípadne aj ďalšie metódy.

0.2 Cieľ práce

Cieľom tejto práce je preskúmať a porovnať rôzne prístupy k optimalizácii trás a dobíjania pre elektromobily s využitím heuristík, ako je ACO a prípadne aj ďalšie algoritmy. Konkrétne sa budeme zameriavať na zlepšenie dojazdu elektromobilov pomocou optimalizácie trás a plánovania dobíjania, s cieľom minimalizovať náklady a čas potrebný na cestovanie, a zároveň maximalizovať ich využitie a efektívnosť.

0.3 Štruktúra práce

Táto práca je organizovaná nasledovne:

- **Kapitola 2:** Teoretické základy elektromobility a optimalizačných metód.
- **Kapitola 3:** Prehľad existujúcich prístupov k optimalizácii trás a dobíjania pre elektromobily.
- **Kapitola 4:** Detailný popis Ant Colony Optimization a ďalších potenciálnych heuristík.
- **Kapitola 5:** Implementácia a experimenty, vrátane porovnania výsledkov rôznych metód.

- **Kapitola 6:** Diskusia a závery, vrátane návrhov budúcich výskumných smerov.

0.4 Prínos práce

Výsledky tejto práce môžu mať dôležité praktické využitie pre odvetvie elektromobility, pretože môžu poskytnúť cenné poznatky a odporúčania pre optimalizáciu trás a dobíjania pre elektromobily v reálnom svete. Okrem toho môžu prispieť k lepšiemu porozumeniu vplyvu heuristik na riešenie optimalizačných úloh v kontexte elektromobility.

Kapitola 1

Teoretické základy elektromobility a optimalizačních metod

Kapitola 2

Prehľad existujúcich prístupov k optimalizácii trás a dobíjania pre elektromobily

[1] [4] [2] [3] [5]

Kapitola 3

Popis Ant Colony Optimization a d'alších potenciálnych heuristik

3.0.1 Úvod

Algoritmy optimalizácie kolóniou mravcov (ACO) sa ukázali ako silné nástroje na riešenie problémov. Majú schopnosť poskytnúť optimálne (alebo takmer optimálne) riešenie pre ťažké problémy s trasovaním vozidiel (VRP) [1]. Tradične sa výskumníci sústredili na statické optimalizačné problémy, kde sa prostredie problému počas optimalizačného procesu algoritmu nezmení. Avšak mnohé reálne aplikácie sú vystavené dynamickým prostrediam [1]. Dynamické optimalizačné problémy (DOPs) sú výzvou, pretože cieľom algoritmu nie je len rýchlo lokalizovať optimum problému, ale aj efektívne sledovať pohybujúce sa optimum pri zmene [1]. Dynamická zmena môže zahŕňať faktory ako objektívna funkcia, vstupné premenné, príklad problému a obmedzenia.

3.0.2 Prispôsobenie sa dynamickým zmenám

Jednoduchý spôsob riešenia DOPs je reštartovať optimalizačný proces algoritmu, keď sa vyskytne dynamická zmena. Avšak táto stratégia sa obvykle používa v prípade, že dynamické zmeny sú vážne. Naopak, keď sú dynamické zmeny malé až stredné, je efektívnejšie prispôbiť sa meniacemu sa prostrediu prenášaním minulých znalostí, pretože nové prostredie bude nejako súvisieť s predchádzajúcim. ACO je dobrá voľba na prispôsobenie sa dynamickým zmenám, pretože prirodzene implementuje pamäťovú štruktúru prostredníctvom feromónových stôp, čo mu umožňuje zapamätať si a prenášať minulé znalosti [1].

3.0.3 Formulácia problému dynamického trasovania vozidiel (DVRP)

VRP je náročný NP-ťažký kombinatorický optimalizačný problém [1]. Problém môže byť popísaný nasledovne: za predpokladu, že máme flotilu vozidiel s obmedzenou kapacitou nákladu, potrebujeme nájsť najlepšiu možnú trasu pre každé vozidlo, začínajúc a končiac na centrálnom sklade, pričom uspokojujeme dodávateľské požiadavky sady zákazníkov [1].

3.0.4 Generovanie dynamických testových prostredí

V práci môžeme použiť dynamický rámec navrhnutý v nedávnej práci [1]. Každý prípad VRP pozostáva z matice váh, ktorá obsahuje vzdialenosti medzi bodmi, a množiny zákazníkov, ktoré sa môžu vložiť alebo odstrániť. Pri vytváraní dynamických zmien sa používajú rôzne typy dynamických zmien, ako sú pridanie a odobratie zákazníkov alebo zmena vzdialeností medzi bodmi.

Kapitola 4

Implementácia a experimenty, vrátane porovnania výsledkov rôznych metód

V tejto kapitole popíšeme implementáciu algoritmu na načítanie benchmarkov pre problém elektrického vozidla s obmedzenou dobíjacou kapacitou (EVRP) a heuristiky generovania riešení. Implementácia je založená na článku [5], kde sú definované parametre benchmarkov a odporúčania pre vyhodnotenie algoritmov.

4.1 Načítanie benchmarkov

Načítanie benchmarkov je kľúčovým prvkom implementácie, pretože zabezpečuje správne načítanie a spracovanie vstupných dát, ktoré sú nevyhnutné pre ďalšie operácie v algoritme. Funkcia `load_data()` je zodpovedná za načítanie dát zo súborov s príponou `.evrp` a ich ukladanie do vhodnej dátovej štruktúry.

Pri implementácii sme sa rozhodli využiť štandardné knižnice na prácu so súborami, aby sme zabezpečili univerzálnosť a jednoduchosť použitia. Okrem toho sme využili špecifické funkcie definované v súbore `EVRP.hpp`, ktoré boli navrhnuté tak, aby spĺňali požiadavky na načítanie špecifických parametrov benchmarkov.

Každý benchmark obsahuje nasledujúce parametre, ktoré sú dôležité pre jeho správne načítanie a spracovanie [5]:

- **Počet zákazníkov (#customers):** Určuje celkový počet zákazníkov, ktorí majú byť obsluhovaní elektromobilmi.
- **Počet depôtov (#depots):** Určuje počet depôtov, z ktorých elektromobily vyštartujú a kam sa vrátia po skončení plánovanej trasy.
- **Počet nabíjacích staníc (#stations):** Určuje počet nabíjacích staníc, na ktorých môžu elektromobily dobíjať svoju batériu.
- **Minimálny počet trás (#routes):** Určuje minimálny počet trás, ktoré musia elektromobily prejsť, aby obslúžili všetkých zákazníkov.

- **Maximálna nosnosť elektromobilu (C):** Určuje maximálnu nosnosť elektromobilu, teda maximálny počet zákazníkov, ktorých môže obslúžiť v rámci jednej trasy.
- **Maximálna kapacita batérie elektromobilu (Q):** Určuje maximálnu kapacitu batérie elektromobilu, teda maximálny počet kilometrov, ktorý je elektromobil schopný prejsť na jedno nabitie batérie.
- **Konštanta spotreby energie (h):** Určuje konštantu spotreby energie, ktorá sa používa na výpočet spotreby energie elektromobilu pri prejení jednotlivých trás.
- **Optimalizovaná horná hranica (UB):** Tento parameter predstavuje optimálnu hornú hranicu, ktorá je buď známa alebo odhadnutá a slúži ako referenčná hodnota pre porovnanie výsledkov algoritmov.

Tieto parametre sú kľúčové pre správne načítanie a spracovanie benchmarkov a zabezpečujú, že algoritmus pracuje s validnými dátami. V ďalšom kroku algoritmus používa tieto dáta na generovanie riešení problému EVRP.

4.2 Heuristika generovania

4.3 Vyhodnotenie

Implementácia načítania benchmarkov a heuristiky generovania umožňuje efektívne testovanie algoritmov pre problém EVRP. Zabezpečuje správne načítanie a spracovanie benchmarkov a generuje kvalitné riešenia s ohľadom na definované parametre a obmedzenia.

Kapitola 5

Diskusia a závery, vrátane návrhov budúcich výskumných smerov

Záver

Literatúra

- [1] Ismael Santana Bonilha, Michalis Mavrovouniotis, Fernando Moreira Muller, Georgios Ellinas, and Marios Polycarpou. Ant colony optimization with heuristic repair for the dynamic vehicle routing problem. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, pages 313–320. IEEE, 2020.
- [2] S. S. Fazeli, S. Venkatachalam, and J. M. Smereka. Efficient algorithms for electric vehicles’ min-max routing problem. *Sustainable Operations and Computers*, pages 0–20, 2023.
- [3] Francisco Lezama, João Soares, Ricardo Faia, Zita Vale, Luís Henrique Macedo, and Raquel Romero. Business models for flexibility of electric vehicles: Evolutionary computation for a successful implementation. In *GECCO 2019 Companion - Proceedings of the 2019 Genetic and Evolutionary Computation Conference Companion*, pages 1873–1878, 2020.
- [4] Michalis Mavrovouniotis, Chao Li, Georgios Ellinas, and Marios Polycarpou. Solving the electric capacitated vehicle routing problem with cargo weight. In *2022 IEEE Congress on Evolutionary Computation, CEC 2022 - Conference Proceedings*, July 2022.
- [5] Michalis Mavrovouniotis, Christos Menelaou, Stelios Timotheou, Christos Panayiotou, Georgios Ellinas, and Marios Polycarpou. Benchmark set for the ieee wcci-2020 competition on evolutionary computation for the electric vehicle routing problem. *Kios Coe*, March 2020.